

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-044089

(43)Date of publication of application : 18.02.1994

(51)Int.Cl.

G06F 9/46

G06F 9/46

G06F 9/38

G06F 9/38

(21)Application number : 05-115913

(71)Applicant : MATSUSHITA ELECTRIC IND  
CO LTD

(22)Date of filing : 18.05.1993

(72)Inventor : KIMURA KOZO  
HIRATA HIROAKI

(30)Priority

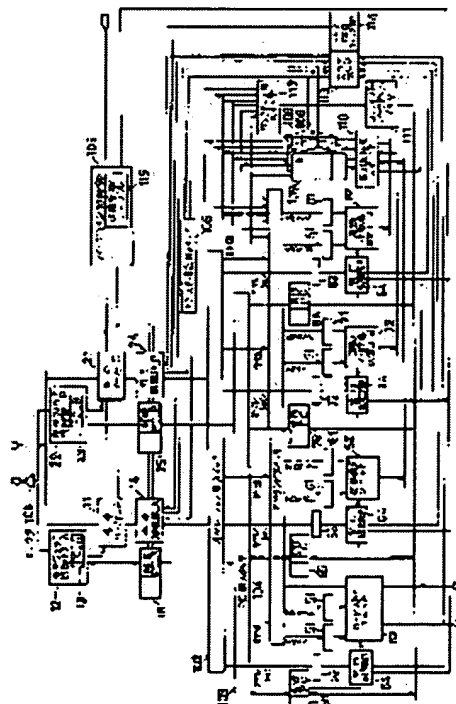
Priority number : 04124910 Priority date : 18.05.1992 Priority country : JP

## (54) INFORMATION PROCESSOR

(57)Abstract:

PURPOSE: To efficiently use the plural arithmetic operation units of plural instruction streams so as to execute them.

CONSTITUTION: Plural operand buffers 51, 61, 71 and 81, the plural arithmetic operation units 52, 62, 72 and 82, plural register files 107-109, execution control parts 54, 64, 74 and 84 controlling the execution of the plural instruction streams and a context back-up memory 111 storing the context of the instruction stream are provided. The execution control parts 54, 69, 74 and 84 stops the execution of the instruction stream of an instruction which becomes a main cause when a specified event



occurs or a specified state is detected, saves the context in the context back-up memory 111 and executes change-over to the new instruction stream context so as to start execution.

---

#### LEGAL STATUS

[Date of request for examination] 17.05.2000

[Date of sending the examiner's decision of rejection] 08.06.2004

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平6-44089

(43) 公開日 平成6年(1994)2月18日

(51) Int.Cl. <sup>5</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/46	3 4 0 Z	8120-5B		
	3 1 3 Z	8120-5B		
9/38	3 1 0 F	9193-5B		
	3 8 0 B	9193-5B		

審査請求 未請求 請求項の数15(全 15 頁)

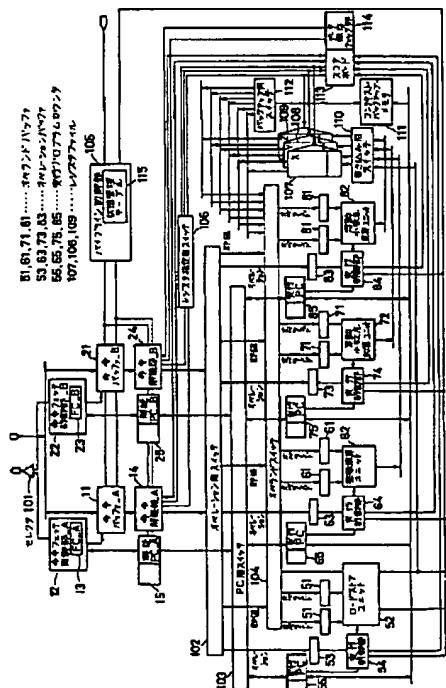
(21) 出願番号	特願平5-115913	(71) 出願人	000005821 松下電器産業株式会社 大阪府門真市大字門真1006番地
(22) 出願日	平成5年(1993)5月18日	(72) 発明者	木村 浩三 大阪府門真市大字門真1006番地 松下電器 産業株式会社内
(31) 優先権主張番号	特願平4-124910	(72) 発明者	平田 博章 大阪府門真市大字門真1006番地 松下電器 産業株式会社内
(32) 優先日	平4(1992)5月18日	(74) 代理人	弁理士 武田 元敏
(33) 優先権主張国	日本 (J P)		

(54) 【発明の名称】 情報処理装置

(57) 【要約】 (修正有)

【目的】 複数の命令ストリームの複数の演算ユニットを効率よく使用して実行する。

【構成】 複数のオペランドバッファ51, 61, 71, 81と、複数の演算ユニット52, 62, 72, 82と、複数のレジスタファイル107~109と、複数の命令流の実行を制御する実行制御部54, 64, 74, 84と、命令流のコンテキストを格納するコンテキスト・バックアップ・メモリ111を備え、実行制御部54, 64, 74, 84は、特定のイベントが発生したり、特定の状態を検出した場合にはその要因となった命令の命令流の実行を停止し、コンテキスト・バックアップ・メモリ111にそのコンテキストを退避し、新たな命令流のコンテキストに切換えて実行を開始する。



1

## 【特許請求の範囲】

【請求項1】 複数の命令準備部と、複数の演算実行部と、複数のレジスタファイルと、前記命令準備部と演算実行部の間に設けられる命令スケジュール部を備え、前記命令準備部は、命令を読み出す命令フェッチ手段と、前記演算実行部からのデータ依存情報を解析、保管する依存解析手段と、前記命令フェッチ手段が読み出した命令を解読して前記データ依存情報を基に命令の発行可能性を判定する命令解読手段からなり、前記命令スケジュール部は、複数の命令準備部からの命令解読結果を受け付けるとともに、複数の演算実行部のうち命令受付可能状態にある演算実行部を対象として発行可能な命令解読結果を選択し、それらの中の各命令を対応する演算実行部に出力する手段を含み、前記演算実行部は、命令スケジュール部から受け付けた命令を実行する手段と、実行終了時には依存解消情報をその命令の解読時に依存解析を行った命令準備部の依存解析手段に通知する手段とを含み、特定のイベントが発生したり、または特定の状態を検出した場合には、その要因となった命令の命令流の実行を停止し、メモリに停止した命令流のコンテキストを退避し、メモリから新たな命令流のコンテキストを格納して実行を開始することを特徴とする情報処理装置。

【請求項2】 複数の命令準備部と、複数の演算実行部と、複数のレジスタファイルと、前記命令準備部と演算実行部の間に設けられる命令スケジュール部と、複数の命令流の実行を制御する命令流実行制御部と、命令流のコンテキストを格納するメモリを備え、前記命令準備部は、命令を読み出す命令フェッチ手段と、前記演算実行部からのデータ依存情報を解析、保管する依存解析手段と、前記命令フェッチ手段が読み出した命令を解読して前記データ依存情報を基に命令の発行可能性を判定する命令解読手段からなり、前記命令スケジュール部は、複数の命令準備部からの命令解読結果を受け付けるとともに、複数の演算実行部のうち命令受付可能状態にある演算実行部を対象として発行可能な命令解読結果を選択し、それらの中の各命令を対応する演算実行部に出力する手段を含み、前記演算実行部は、命令スケジュール部から受け付けた命令を実行する手段と、実行終了時には依存解消情報をその命令の解読時に依存解析を行った命令準備部の依存解析手段に通知する手段とを含み、前記命令流実行制御部は、特定のイベントが発生したり、または特定の状態を検出した場合には、その要因となった命令の命令流の実行を停止し、メモリに停止した命令流のコンテキストを退避し、メモリから新たな命令流のコンテキストを格納して実行を開始することを特徴とする情報処理装置。

【請求項3】 複数の命令準備部と、複数の演算実行部と、少なくとも前記命令準備部よりも多くの複数のレジスタファイルと、前記命令準備部と演算実行部の間に設けられる命令スケジュール部と、複数の命令流の実行を

2

制御する命令流実行制御部を備え、前記命令準備部は、命令を読み出す命令フェッチ手段と、前記演算実行部からのデータ依存情報を解析、保管する依存解析手段と、前記命令フェッチ手段が読み出した命令を解読して前記データ依存情報を基に命令の発行可能性を判定する命令解読手段からなり、前記命令スケジュール部は、複数の命令準備部からの命令解読結果を受け付けるとともに、複数の演算実行部のうち命令受付可能状態にある演算実行部を対象として発行可能な命令解読結果を選択し、それらの中の各命令を対応する演算実行部に出力する手段を含み、前記演算実行部は、命令スケジュール部から受け付けた命令を実行する手段と、実行終了時には依存解消情報をその命令の解読時に依存解析を行った命令準備部の依存解析手段に通知する手段とを含み、前記命令流実行制御部は、命令流を命令準備部およびレジスタファイルに対応させて、命令準備部に割り付けられている命令流については実行させるが、特定のイベントが発生したり、または特定の状態を検出した場合には、その要因となった命令の命令流の実行を停止し、その命令流に割り付けられていた命令準備部をまだ割り付けられていなかった命令流に対応させて、新たに割り付けられた命令流の実行を開始することを特徴とする情報処理装置。

【請求項4】 複数の命令準備部と、複数の演算実行部と、少なくとも前記命令準備部よりも多くの複数のレジスタファイルと、前記命令準備部と演算実行部の間に設けられる命令スケジュール部と、複数の命令流の実行を制御する命令流実行制御部と、命令流のコンテキストを格納するメモリを備え、前記命令準備部は、命令を読み出す命令フェッチ手段と、前記演算実行部からのデータ依存情報を解析、保管する依存解析手段と、前記命令フェッチ手段が読み出した命令を解読して前記データ依存情報を基に命令の発行可能性を判定する命令解読手段からなり、前記命令スケジュール部は、複数の命令準備部からの命令解読結果を受け付けるとともに、複数の演算実行部のうち命令受付可能状態にある演算実行部を対象として発行可能な命令解読結果を選択し、それらの中の各命令を対応する演算実行部に出力する手段を含み、前記演算実行部は、命令スケジュール部から受け付けた命令を実行する手段と、実行終了時には依存解消情報をその命令の解読時に依存解析を行った命令準備部の依存解析手段に通知する手段とを含み、前記命令流実行制御部は、命令流を命令準備部およびレジスタファイルに対応させて、命令準備部に割り付けられている命令流については実行させるが、特定のイベントが発生したり、または特定の状態を検出した場合には、その要因となった命令の命令流の実行を停止し、その命令流のコンテキストをメモリに退避し、その命令流に割り付けられていた命令準備部をまだ命令準備部に割り付けられておらず、かつレジスタファイルには割り付けられている命令流に対応させて、新たに割り付けられた命令流の実行を開始す

ることを特徴とする情報処理装置。

【請求項5】 命令流の切り替えの契機となる特定のイベントをキャッシュアクセス時のミスヒットとすることを特徴とする請求項1ないし4のいずれかに記載の情報処理装置。

【請求項6】 特定のイベントが発生したり、または特定の状態を検出した場合には、その要因となった命令の命令流に割り付けられている命令準備部の実行を停止し、命令準備部の内部状態を初期化することを特徴とする請求項1ないし4のいずれかに記載の情報処理装置。

【請求項7】 命令流の実行を停止し退避するコンテキストに先頭PCを含み、再度この命令流を実行する場合には先頭PCから実行を再開することを特徴とする請求項1ないし4のいずれかに記載の情報処理装置。

【請求項8】 命令流の切り替えの契機となる特定のイベントをキャッシュアクセス時のミスヒットとし、キャッシュミスが発生した場合、メモリアクセス命令を打ち切り正常終了させることを特徴とする請求項1ないし4のいずれかに記載の情報処理装置。

【請求項9】 新たな命令流を命令準備部に割り付け、命令実行を再開する場合、切り替えの要因となり実行を打ち切ったメモリアクセス命令を再実行させることを特徴とする請求項1ないし4のいずれかに記載の情報処理装置。

【請求項10】 命令流の切り替えの契機となる特定のイベントをキャッシュアクセス時のミスヒットとし、命令流のコンテキストを退避する場合にはメモリアクセス命令のPCも一緒に退避することを特徴とする請求項1ないし4のいずれかに記載の情報処理装置。

【請求項11】 命令流の切り替えの契機となる特定のイベントをキャッシュアクセス時のミスヒットとし、命令流のコンテキストを退避する場合にはメモリアクセス命令も一緒に退避することを特徴とする請求項1ないし4のいずれかに記載の情報処理装置。

【請求項12】 コンテキストを退避するメモリが同一プロセッサ上にあることを特徴とする請求項2または4記載の情報処理装置。

【請求項13】 命令流実行制御部は、コンテキストがメモリに退避された命令流については、特定の状態が解消され、かつ命令流に割り付けられていないレジスタファイルがある場合、コンテキストがレジスタファイルに復元され命令流がそのレジスタファイルに割り付けることを特徴とする請求項4記載の情報処理装置。

【請求項14】 命令流の切り替えの契機となる特定のイベントをキャッシュアクセス時のミスヒットとし、命令流のコンテキストを退避する場合にはメモリアクセスアドレスも一緒に退避することを特徴とする請求項1ないし4のいずれかに記載の情報処理装置。

【請求項15】 命令流の切り替えの契機となる特定のイベントをメモリアクセスに関するエラーとすることを

特徴とする請求項1ないし4のいずれかに記載の情報処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は複数の命令ストリームの命令を並列に発行することによって、複数の演算ユニットを効率よく使用する情報処理装置に関する。

【0002】

【従来の技術】従来の情報処理装置の例としては、一つのプロセッサ内で複数の命令流を同時に処理するマルチスレッド・プロセッサがある。この方式については、“A Multi threaded Processor Architecture with Simultaneous Instruction Issuing,” In Proc. of ISS'91: International Symposium on Supercomputing, Fukuoka, Japan, pp.87-96, November 1991 に詳細に述べられている。この従来の情報処理装置の構成図を図7に示す。図7において200は命令キャッシュ、201は命令フェッチユニット、202は解説ユニット、203はスタンバイステーション、204は命令スケジュールユニット、205は機能ユニット、206はレジスタセットである。以上のように構成された従来例の情報処理装置について、その動作を説明する。

【0003】各命令フェッチユニット201はそれぞれ異なる命令流の命令を命令キャッシュ200から読み込む。解説ユニット202はそれぞれの命令流の命令を解説し、命令を処理可能な機能ユニット205に接続されているスタンバイステーション203に格納する。命令スケジュールユニット204はスタンバイステーション203から適切な命令を選択し、機能ユニット205に送る。機能ユニット205はレジスタセット206を用いて実行する。

【0004】このプロセッサの特徴は複数の命令流を演算器で共有して実行することである。既存のスーパースカラ処理方式のプロセッサは機能ユニット205のみの多重化(複数化)のため、同時に処理可能な命令ストリームは1つで、命令間の依存関係によりパイプラインインタロックが頻繁に発生する。その結果、機能ユニット205の使用効率は上がり性能向上が困難であった。また、従来例のプロセッサでは複数の命令ストリームの命令を並列に実行することにより命令レベルの並列性を増加し、各機能ユニットの使用効率を上げ、性能向上を実現できる。

【0005】

【発明が解決しようとする課題】しかしながら上記の構成では、下記の問題点を有していた。機能ユニット205の1つのロードストアユニットにおいて、ある1つの命令ストリームのロード命令を実行した時、キャッシュミスを起こしたとする。キャッシュミスが発生するとキャッシュは主記憶よりデータのアップデートを行うためロードストアユニットはインタロック状態に陥る。これより、長時間にわたり、キャッシュミスを起こした命令ス

5

トリームの解読ユニット202などが命令発行待ちの状態となり、ロード命令に後続する命令が実行できなくなる。そして、他の命令ストリーム上の命令についてもロードストアユニットを使用できなくなり、やがてプロセッサ全体が停止してしまうことになる。このように、ある1箇所でパイプラインのインタロックが発生するとプロセッサ全体に影響を及ぼし、性能劣化を引き起こす。

【0006】本発明は上記問題点に鑑み、複数の命令ストリームを同時実行するプロセッサにおいて、パイプラインのインタロックおよびロック時間を削減し、高性能を実現する情報処理装置の提供を目的とする。

【0007】

【課題を解決するための手段】本発明は、上記問題点を解決するために、請求項1記載の発明の情報処理装置は、複数の命令準備部と、複数の演算実行部と、複数のレジスタファイルと、前記命令準備部と演算実行部の間に設けられる命令スケジュール部を備え、特定のイベントが発生したり、または特定の状態を検出した場合には、その要因となった命令の命令流の実行を停止し、メモリに停止した命令流のコンテキストを退避し、メモリから新たな命令流のコンテキストを格納して実行を開始することを特徴としている。

【0008】請求項2記載の発明の情報処理装置は、複数の命令準備部と、複数の演算実行部と、複数のレジスタファイルと、前記命令準備部と演算実行部の間に設けられる命令スケジュール部と、複数の命令流の実行を制御する命令流実行制御部と、命令流のコンテキストを格納するメモリを備え、命令流実行制御部は、特定のイベントが発生したり、または特定の状態を検出した場合には、その要因となった命令の命令流の実行を停止し、メモリに停止した命令流のコンテキストを退避し、メモリから新たな命令流のコンテキストを格納して実行を開始することを特徴としている。

【0009】請求項3記載の発明の情報処理装置は、複数の命令準備部と、複数の演算実行部と、少なくとも前記命令準備部よりも多くの複数のレジスタファイルと、前記命令準備部と演算実行部の間に設けられる命令スケジュール部と、複数の命令流の実行を制御する命令流実行制御部を備え、前記命令流実行制御部は、命令流を命令準備部およびレジスタファイルに対応させて、命令準備部に割り付けられている命令流については実行させるが、特定のイベントが発生したり、または特定の状態を検出した場合には、その要因となった命令の命令流の実行を停止し、その命令流に割り付けられていた命令準備部をまだ割り付けられていなかった命令流に対応させて、新たに割り付けられた命令流の実行を開始することを特徴としている。

【0010】請求項4記載の発明の情報処理装置は、複数の命令準備部と、複数の演算実行部と、少なくとも前記命令準備部よりも多くの複数のレジスタファイルと、

6

前記命令準備部と演算実行部の間に設けられる命令スケジュール部と、複数の命令流の実行を制御する命令流実行制御部と、命令流のコンテキストを格納するメモリを備え、前記命令流実行制御部は、命令流を命令準備部およびレジスタファイルに対応させて、命令準備部に割り付けられている命令流については実行させるが、特定のイベントが発生したり、または特定の状態を検出した場合には、その要因となった命令の命令流の実行を停止し、その命令流に割り付けられていた命令準備部をまだ割り付けられていなかった命令流に対応させて、新たに割り付けられた命令流の実行を開始することを特徴としている。

【0011】請求項5記載の発明の情報処理装置は、命令流の切り替えの契機となる特定のイベントをキャッシュアクセス時のミスヒットとすることを特徴としている。

【0012】請求項6記載の発明の情報処理装置は、特定のイベントが発生したり、または特定の状態を検出した場合には、その要因となった命令の命令流に割り付けられている命令準備部の実行を停止し、命令準備部の内部状態を初期化することを特徴としている。

【0013】請求項7記載の発明の情報処理装置は、命令流の実行を停止し退避するコンテキストに先頭PCを含み、再度この命令流を実行する場合には先頭PCから実行を再開することを特徴としている。

【0014】請求項8記載の発明の情報処理装置は、命令流の切り替えの契機となる特定のイベントをキャッシュアクセス時のミスヒットとし、キャッシュミスが発生した場合、メモリアクセス命令を打ち切り正常終了させることを特徴としている。

【0015】請求項9記載の発明の情報処理装置は、新たな命令流を命令準備部に割り付け、命令実行を再開する場合、切り替えの要因となり実行を打ち切ったメモリアクセス命令を再実行させることを特徴としている。

【0016】請求項10記載の発明の情報処理装置は、命令流の切り替えの契機となる特定のイベントをキャッシュアクセス時のミスヒットとし、命令流のコンテキストを退避する場合にはメモリアクセス命令のPCも一緒に退避することを特徴としている。

【0017】請求項11記載の発明の情報処理装置は、命令流の切り替えの契機となる特定のイベントをキャッシュアクセス時のミスヒットとし、命令流のコンテキストを退避する場合にはメモリアクセス命令も一緒に退避することを特徴としている。

【0018】請求項12記載の発明の情報処理装置は、コンテキストを退避するメモリが同一プロセッサ上にあることを特徴としている。

【0019】請求項13記載の発明の情報処理装置は、命令流の実行制御部は、コンテキストがメモリに退避された命令流については、特定の状態が解消され、かつ命令

流に割り付けられていないレジスタファイルがある場合、コンテキストがレジスタファイルに復元され命令流がそのレジスタファイルに割り付けることを特徴としている。

【0020】請求項14記載の発明の情報処理装置は、命令流の切り替えの契機となる特定のイベントをキャッシュアクセス時のミスヒットとし、命令流のコンテキストを退避する場合にはメモリアクセスアドレスも一緒に退避することを特徴としている。

【0021】請求項15記載の発明の情報処理装置は、命令流の切り替えの契機となる特定のイベントをメモリアクセスに関するエラーとすることを特徴としている。

【0022】

【作用】請求項1記載の発明の情報処理装置においては、インタロックが発生した命令流を休眠状態(退避)にし、ハードウェア機構を他の命令流に割り付ける、すなわち命令流を切り替えて実行することにより、パイプラインのインタロックおよびロック時間を削減し、機能ユニットの使用効率を向上させ、高性能を実現できる。

【0023】請求項2記載の発明の情報処理装置においては、インタロックが発生した命令流を休眠状態にしてコンテキストをメモリに退避し、新たな命令流のコンテキストを復帰させ、ハードウェア機構を割り付ける、すなわち命令流を切り替えて実行することにより、パイプラインのインタロックおよびロック時間を削減し、機能ユニットの使用効率を向上させ、高性能を実現できる。

【0024】請求項3記載の発明の情報処理装置においては、命令準備部よりも多くのレジスタファイルを設けることにより、インタロックが発生しても、インタロックが発生した命令流のコンテキストをメモリに退避することなく、新たな命令流を実行させることができるので、高速な命令流切り替えができる。

【0025】請求項4および13記載の発明の情報処理装置においては、命令準備部よりも多くのレジスタファイルとコンテキストを退避するメモリを設けることにより、インタロックが発生した場合、インタロックが発生した命令流のコンテキストはメモリに退避するが、新たな命令流のコンテキストは予めレジスタに準備することができるので、通常の命令と命令流の切り替えをオーバーラップして実行できるので、高速な命令流切り替えができる。

【0026】請求項5記載の発明の情報処理装置においては、命令流の切り替えの契機となる特定のイベントをキャッシュアクセス時のミスヒットとすることにより、メモリアクセス時のオーバーヘッドを削減できる。

【0027】請求項6記載の発明の情報処理装置においては、特定のイベントが発生したり、または特定の状態を検出した場合には、その要因となった命令の命令流に割り付けられている命令準備部の実行を停止し、命令準備部の内部状態を初期化することにより、パイプライン

制御が容易になる。

【0028】請求項7記載の発明の情報処理装置においては、命令流の実行を停止し退避するコンテキストに先頭PCを含み、再度この命令流を実行する場合には先頭PCから実行を再開することにより、命令切り替えの制御が容易になる。

【0029】請求項8記載の発明の情報処理装置においては、命令流の切り替えの契機となる特定のイベントをキャッシュアクセス時のミスヒットとし、キャッシュミスが発生した場合、メモリアクセス命令を打ち切り正常終了させることにより、命令切り替えの制御が容易になる。

【0030】請求項9記載の発明の情報処理装置においては、新たな命令流を命令準備部に割り付け、命令実行を再開する場合、切り替えの要因となり実行を打ち切ったメモリアクセス命令を再実行させることにより、命令切り替えの制御が容易になる。

【0031】請求項10記載の発明の情報処理装置においては、命令流の切り替えの契機となる特定のイベントをキャッシュアクセス時のミスヒットとし、命令流のコンテキストを退避する場合にはメモリアクセス命令のPCも一緒に退避することにより、命令再実行が可能となり切り替えの制御が容易になる。

【0032】請求項11記載の発明の情報処理装置においては、命令流の切り替えの契機となる特定のイベントをキャッシュアクセス時のミスヒットとし、命令流のコンテキストを退避する場合にはメモリアクセス命令も一緒に退避することにより、命令再実行が高速に実現でき切り替えの制御も容易になる。

【0033】請求項12記載の発明の情報処理装置においては、コンテキストを退避するメモリを同一プロセッサ上に実現することにより、高速な切り替えやロード命令などの外部バスを使用する命令と並列に実行できる。

【0034】請求項14記載の発明の情報処理装置においては、命令流の切り替えの契機となる特定のイベントをキャッシュアクセス時のミスヒットとし、命令流のコンテキストを退避する場合にはメモリアクセスアドレスも一緒に退避することにより、命令再実行が高速に実現でき切り替えの制御も容易になる。

【0035】請求項15記載の発明の情報処理装置においては、命令流の切り替えの契機となる特定のイベントをメモリアクセスに関するエラーとすることにより、メモリアクセスエラー時のオーバーヘッドを削減できる。

【0036】

【実施例】図1は本発明の一実施例における情報処理装置の構成図を示すものである。図1において、11および21はメモリからフェッチした命令を格納しておく命令バッファ、12および22はフェッチしてくる命令のアドレス計算およびアドレス(PC(プログラムカウンタ)値と同じになる)管理を行う命令フェッチ制御部、13および23

はフェッチする命令のアドレスを格納する命令フェッチカウンタ(以下、FCと略す)、14および24は命令バッファ11および21から出力する命令を解読する命令解読部、15および25は命令解読部14および24と対応して解読中の命令のアドレスを格納する解読プログラムカウンタ(以下、解読PCと略す)であり、この解読PC\_A15および解読PC\_B25は解読PC値だけではなく命令流のID番号を格納している。

【0037】51, 61, 71および81はレジスタからの命令を一時的に格納するオペランドバッファ、52, 62, 72および82は演算ユニットであり、この演算ユニットの機能は特に限定する必要はないが説明を簡単にするため、かつ一般的な構成を考えると、52はメモリアクセス命令を処理するロードストアユニット、62は整数演算を処理する整数演算ユニット、72は浮動小数点の加減算や整数と浮動小数点間の変換を行う浮動小数点加算ユニット、82は浮動小数点の乗算や除算を行う浮動小数点乗算ユニットである。

【0038】53, 63, 73および83はオペレーションまたは命令を一時的に格納するオペレーションバッファ、54, 64, 74および84は各演算ユニットを制御する実行制御部、55, 65, 75および85は実行中の命令のアドレス(実行PC)を格納する実行プログラムカウンタ(以下、実行PCと略す)であり、この実行PC\_55および実行PC\_65は実行PC値だけではなく命令流のID番号を格納している。

【0039】101は命令フェッチ用のアドレスを命令フェッチ制御部12および22から選択するセレクタ、102は命令の解読結果であるオペレーションおよび即値などを処理対象の演算ユニットに接続するオペレーション用スイッチ、103はオペレーション用スイッチ102と連動して解読PC値を対応する実行PCに接続するPC用スイッチ、104は命令解読部14および24から出力された即値とレジスタファイル107, 108および109から出力されたオペランドをオペランドバッファ51, 61, 71および81に接続するオペランドスイッチである。

【0040】105はパイプラインを無効化したり、命令を再実行させたりなどのパイプライン制御を司るパイプライン制御部、106は実行対象となっているレジスタファイル107, 108, 109を選択してレジスタ番号を出力するレジスタ指定用スイッチ、前記レジスタファイル107, 108および109は演算ユニット52, 62, 72, 82に対応して整数用と浮動小数点用レジスタを持つ。

【0041】110は演算結果を対応するレジスタに接続する書き込み用スイッチ、111は命令流を切り替えたときにレジスタファイル107, 108および109の内容や実行PC値を格納するコンテキスト・バックアップ・メモリ、112はレジスタファイル107, 108および109とコンテキスト・バックアップ・メモリ111間でデータを転送する場合に両者を接続するバックアップ用スイッチ、113

は命令実行中でデータが確定していないレジスタの番号を示すスコアボード、114はスコアボードの状態と命令よりデータ依存が発生しているか否かを調べるデータ依存チェック部、115は命令流の状態と対応する命令フェッチユニットと対応するレジスタファイルを示す状態管理テーブルである。

【0042】説明を分かりやすくするために、同じ機能を持つ機構(命令バッファや命令解読部など)については、図1に示すように名称の最後に適当なアルファベットの付加して区別する。同じアルファベットが付加されたものは、同一の命令流を扱うと考えてよい。

【0043】命令バッファ11および21をそれぞれ命令バッファ\_A、命令バッファ\_Bとし、同様に命令解読部14および24もそれぞれ命令解読部\_A、命令解読部\_Bとする。命令フェッチ制御部12および22や解読PC15および25等についても図1に示すとおり同様である。レジスタファイル107, 108および109については、それぞれレジスタファイル\_X、レジスタファイル\_Y、レジスタファイル\_Zとする。

【0044】以上のように構成された本実施例の情報処理装置について、以下図1を用いてその動作を説明する。まず、動作を説明する前に、データ(命令も含む)や構成に関する前提条件について述べる。命令流(スレッド)は4個とし説明を分かりやすくするために、命令流1、命令流2、命令流3、命令流4とする。図1に示したようにレジスタファイルはX, Y, Zの3セット(107, 108, 109)とする。同様に、命令フェッチユニット(命令バッファ11, 21および命令フェッチ制御部12, 22などを含む)や、命令解読部14, 24などは2セットとする。従って、同時実行の可能な命令流は2個となる。演算ユニット52, 62, 72, 82はパイプライン化されているが、パイプライン段数を始め詳細な構成等については本発明とは直接関係しないので特に規定しない。また、命令解読部14, 24は命令流1個につき1命令を解読し、一度に発行できる命令も1個とする。ロードストアユニット52はキャッシュユニットに接続され、ヒット時には高速にデータをアクセスできるとする。

【0045】命令流の状態と対応する命令フェッチユニットおよびレジスタファイルの初期状態を表1に示す。

【0046】

【表1】

	状 態	レジスタ	命令フェッチ ユニット
命令流1	実 行	X	A
命令流2	実 行	Y	B
命令流3	可 能	Z	なし
命令流4	可 能	メモリ	なし

【0047】これはパイプライン制御部105内の状態管理テーブル115の内容と等価であり、パイプライン制御



11

部105はこの状態管理テーブル115の内容を更新すると同時に、状態管理テーブル115の内容に基づいて命令流やパイプラインを制御する。表1を用いて初期状態を説明する。

【0048】命令流1および2が情報処理装置内部で現在実行中であり、命令流3および4は実行可能状態にある。命令流1および2は実行状態にあるのでレジスタファイル107, 108, 109に割り付けられており、命令フェッチユニットは命令流1および2の命令をフェッチしている。従って、命令流1については、命令フェッチユニット\_A(命令バッファ\_A11や命令フェッチ制御部\_A12等を含む)で命令をフェッチし、命令解読部\_A14(解読PC\_A15等も含む)で命令を解読し、演算は演算ユニット(演算ユニットについては命令流に関わらず命令の種類によって使用する)でレジスタファイル\_X107を使用して実行する。

【0049】同様に、命令流2については、命令フェッチユニット\_B(命令バッファ\_B21や命令フェッチ制御部\_B22等を含む)で命令をフェッチし、命令解読部\_B24(解読PC\_B25等も含む)で命令を解読し、演算は演算ユニット(演算ユニットについては命令流に関わらず命令の種類によって使用する)でレジスタファイル\_Y108を使用して実行する。命令流3および命令流4は、初期状態は実行可能状態なので、命令フェッチバッファや命令解読部等は使用できない。しかし、レジスタファイルはもう1セット設けてあるので、命令流3をレジスタファイル\_Z109に割り付けることにする。

【0050】演算ユニットでは命令流1または命令流2の命令が実行されており、処理する命令がない場合にはアイドル状態になっている。スコアボード113では実行中により結果が確定していないレジスタ番号が命令流ごとに管理されている。コンテキスト・バックアップ・メモリ111には現在レジスタファイル107, 108, 109に格納されていない命令流のレジスタ資源などが格納されている。

【0051】次に切り替えが発生しない定常状態についての動作を説明する。

【0052】命令フェッチ制御部\_A12は命令流1の命令をフェッチし命令バッファ\_A11に格納している。同様に、命令フェッチ制御部\_B22は命令流2の命令をフェッチし命令バッファ\_B21に格納している。命令フェッチ制御部\_A12および\_B22は切れ目なく命令を供給するのが主機能であるから、その他の構成とは独立に命令バッファ\_A11および\_B21に命令を供給する。命令バッファ\_A11および\_B21がフル(一杯)になれば命令フェッチを停止する。FC\_A13および\_B23は命令をフェッチする際の命令のアドレスを格納している。定常状態ではアドレスをインクリメントしてフェッチ先のアドレスを計算しておく。また、フェッチ先のアドレスはセレクト101で選択される。分岐が発生した場合には分岐

12

先命令のアドレス計算や分岐先命令のフェッチなどの動作もあるが、本発明とは特に関係しないので省略する。

【0053】命令解読部\_A14および\_B24より以下のステージの説明は、動作を分かり易くするために命令流1および命令流2の命令から動作を説明する。

【0054】命令解読部\_A14は命令バッファ\_A11からLOAD命令(メモリからレジスタへのロード命令)を、命令解読部\_B24は命令バッファ\_B21からADD命令(整数レジスタ間の加算命令)を取り出し、それぞれ解読しオペレーションを作成するとともに、そのオペレーションを処理すべき演算ユニットを決定する。同時に同一命令流内でデータ依存関係が発生していないかをチェックする。依存関係のチェックは、スコアボード113とデータ依存チェック部114が行う。

【0055】具体的には現在実行中のためにレジスタの値が確定していないレジスタ番号がスコアボード113に登録されており、命令解読部からこれから読み出すレジスタ番号と比較し、一致すればデータ依存発生を命令解読部に返す。これから実行する命令が、結果を反映していないレジスタの値を使用することを防ぐためである。レジスタ番号の登録は命令解読部\_A14および\_B24が命令を演算ユニットに発行するときに登録し、レジスタ番号の解除は各演算ユニットの実行制御部54, 64, 74および84が命令実行の終了とともに行う。

【0056】解読PC\_A15と解読PC\_B25は命令解読部\_A14および命令解読部\_B24に対応するPC値を格納しているので、命令解読部\_A14, \_B24が命令バッファ\_A11, \_B21より命令を受け取ったときには、同時に命令フェッチ制御部\_A12, \_B22よりPC値を受け取る。前述したように解読PC\_A15には解読PC値以外に命令流1のIDが、解読PC\_B25には命令流2のIDが格納されている。データ依存関係については信学技報CPSY-90-54('90.7)「SIMP(単一命令流/多重命令パイプライン)方式に基づくスーパースカラ・プロセッサの改良方針」に詳細に解説されている。

【0057】データ依存が発生している場合には依存関係が解除されるまで命令の発行を停止し、発生していない場合には、命令解読部はオペレーションを演算ユニットに発行すると同時に、レジスタファイルへ読み出し要求を出す。命令流1はレジスタファイル\_X107を使用しているので、命令解読部\_A14はレジスタファイル\_X107へレジスタ番号を送出し、命令解読部\_B24はレジスタファイル\_Y108へ送出する。同時に、解読PC\_A15および解読PC\_B25の解読PC値を処理される演算ユニットに送出する。レジスタ指定用スイッチ106は状態管理テーブル115を基に命令解読部\_A14とレジスタファイル\_X107を、命令解読部\_B24とレジスタファイル\_Y108を接続する。オペレーション用スイッチ102は解読結果より命令解読部と演算ユニットを接続する。同様にPC用スイッチ103は解読結果より解読PC

13

と実行PCを接続する。本実施例の場合は、命令解説部\_A14からLOAD命令が、命令解説部\_B24からADD命令が発行されるので、オペレーション用スイッチ102は命令解説部\_A14とロードストアユニット52、命令解説部\_B24と整数演算ユニット62を接続する。PC用スイッチ103も同様の対応関係で接続(解説PC\_A15と実行PC55、解説PC\_B25と実行PC65)を行う。命令流1はレジスタファイル\_X107を使用し、命令流2はレジスタファイル\_Y108を使用しているので、オペランドスイッチ104はレジスタファイル\_X107とロードストアユニット52を、レジスタファイル\_Y108と整数演算ユニット62を接続する。

【0058】ロードストアユニット52の入力にあるオペランドバッファ51にはレジスタファイル\_X107から読み出したオペランド値を、整数演算ユニット62の入力にあるオペランドバッファ61にはレジスタファイル\_Y108から読み出したオペランド値を格納する。そして命令解説部\_A14が発行したオペレーションをオペレーションバッファ53に、命令解説部\_B24が発行したオペレーションをオペレーションバッファ63に、解説PC\_A15の解説PC値および命令流1のIDを実行PC\_A55に、解説PC\_B25の解説PC値および命令流2のIDを実行PC\_B65にそれぞれ格納する。

【0059】以下、各演算ユニットはオペレーションやオペランドを基に実行し、結果をそれぞれレジスタやメモリ等に格納する。ロードストアユニット52においては、命令流1のLOAD命令が実行され、計算したメモリアドレスを基にキャッシュからデータをフェッチし、レジスタファイル\_X107内のレジスタに格納する。整数演算ユニット62においてはADD命令が実行され、加算結果をレジスタファイル\_Y108内のレジスタに格納する。各実行制御部54、64はロードストアユニット52とレジスタファイル\_X107を、整数演算ユニット62とレジスタファイル\_Y108を接続するように書き込み用スイッチ110を制御する。

【0060】実行制御部54、64は演算が終了しレジスタファイル\_X107、\_Y108への書き込みが終了すると、スコアボード113に登録されたレジスタ番号をクリアする。同時に、実行PCを無効化する(後続する命令が入ってくれば必然的に更新される)。なお、演算ユニットがパイプライン構成されているので、実行PCもそれに対応して複数段必要であるが、本発明とは直接関係ないので省略する。

【0061】また、命令解説部から発行されたオペレーションが同じ演算ユニットを使用する場合、本実施例の構成ではどちらか一方のオペレーションを待たせるための機構が必要になるが、本発明とは直接関係ないのでその機構は省略する。演算ユニットの構成が変わった場合には待ち合わせ機構も不要になる可能性もある。同様に、同じ命令流の2つの命令が同時に演算が終了した場

14

合には、レジスタファイルへの書き込みを待たせる、またはレジスタファイルの書き込みポートを複数設けるなどの対策が必要であるが、本発明とは直接関係ないのでその説明は省略する。

【0062】続いて、命令流の切り替えが発生する場合について、以下図2に示す命令流の状態遷移図と、図3に示す命令流が実行状態から休眠状態へ遷移する説明図と、図4に示す命令流が休眠状態から実行可能状態へ遷移する説明図と、図5に示す命令流が実行可能状態から実行状態へ遷移する説明図と、図6に示すコンテキスト・バックアップ・メモリのデータ配置図を用いて動作を説明する。

【0063】図2は命令流の状態と状態遷移をさせるイベントの関係を示した状態遷移図である。命令流1は実行中の状態(ア)なのでプロセッサの資源を使用して命令を処理している。ロードストアユニット52で命令流1のロード命令を実行するが、キャッシュミスを起こすとする。キャッシュミスが発生するとキャッシュは主記憶よりデータのアップデートを行うためロードストアユニット52はインタロック状態に陥る。そこで他の命令流を実行状態にするため命令流1を休眠状態(イ)にする。命令流1はキャッシュが主記憶からデータをアップデートすれば実行させてもかまわない。そこで実行可能状態(ウ)に遷移させる。命令流1は実行可能状態なので、実行のためのハードウェア資源が確保できれば実行状態(ア)に遷移できる。

【0064】命令流の切り替えが発生する場合について、次に概説するような命令流の状態遷移に沿って説明する。

【0065】(1) 命令流1および命令流2は実行中の状態なのでプロセッサの資源を使用して(命令流1がレジスタファイル\_X107、命令流2がレジスタファイル\_Y108)命令を処理している。(初期状態、命令流1および命令流2：実行状態)

(2) ロードストアユニット52で命令流1のロード命令を実行するが、キャッシュミスを発生する。

【0066】(3) 命令流1のコンテキストをコンテキスト・バックアップ・メモリ111に退避し、命令流1を休眠状態にする。(命令流1：実行状態→休眠状態)

(4) 実行可能状態でレジスタファイルにコンテキストが用意されている命令流3を実行するためにFC値などを設定する。

【0067】(5) 命令流3に起動をかける。(命令流3：実行可能状態→実行状態)

(6) 命令流4は実行可能状態なのでコンテキスト・バックアップ・メモリ111からコンテキストをレジスタファイル\_X107に格納する。

【0068】(7) キャッシュはミスヒットしたデータをアップデートする。(命令流1：休眠状態→実行可能状態)

上記(1)~(7)に沿って詳細な動作を説明する。

【0069】(1-1) 命令流1のLOAD命令がロードストアユニット52においてキャッシュアクセスするまで上述した定常状態の説明と同じである。命令流の初期状態については前出の表1に示す。また、一連の動作の様子については図3に示す。

【0070】(2-1) キャッシュアクセスの結果、キャッシュミスを起こす。キャッシュミス情報がパイプライン制御部105に送られる。

【0071】(2-2) パイプライン制御部105はロードストアユニット52の実行制御部54にキャッシュミスが発生した命令流の調査を依頼する。実行制御部54は命令流1であることをパイプライン制御部105に伝える。

【0072】(2-3) パイプライン制御部105は、状態管理テーブル115を参照し、命令流1の命令がキャッシュミスヒットを起こしたことにより、命令解説部\_A14、命令バッファ\_A11、命令フェッチ制御部\_A12、解説PC\_A15など命令解説部より上流部の処理を停止させ、命令の発行を禁止する。

【0073】(2-4) パイプライン制御部105は、実行制御部54にロードストアユニット52を正常終了させるよう制御する。ただし、実際の書き込みレジスタには何も書き込まずに終了させる。

【0074】(2-5) パイプライン制御部105は、実行制御部64、74および84を経由して、命令流1の命令の処理状況を調査し、すでに演算ユニットで処理中のものがあれば終了まで停止する。

【0075】(3-1) パイプライン制御部105は、すでに演算ユニットで処理中の命令流1の命令がなくなれば、命令流1のコンテキストをコンテキスト・バックアップ・メモリ111に退避する。コンテキストは本実施例の場合にはレジスタファイル\_X107、解説PC\_A15(命令流のIDも含む)、ロードストアユニット52の実行PC\_A55(LOAD命令のPC)、ロードストアユニット52のオペレーションバッファ53(LOAD命令のオペレーション)およびメモリアクセスアドレスである。通常、メモリアクセスアドレスは、オペランドバッファ51の内容を使用して計算する。コンテキスト・バックアップ・メモリ111に格納された命令流の実行を停止し退避するコンテキストの様子を図6に示し、上記命令流を実行する場合にはコンテキストに先頭PCを含み、この先頭PCから実行する。

【0076】(3-2) パイプライン制御部105は命令流1に関して有為なものがないようにスコアボード113等も無効化する。

【0077】(3-3) パイプライン制御部105は、状態管理テーブル115の命令流1の状態を実行から休眠状態へ遷移させる。これを表2に示す。

【0078】

【表2】

	状態	レジスタ	命令フェッチユニット
命令流1	休眠	メモリ	A
命令流2	実行	Y	B
命令流3	実行	Z	なし
命令流4	可能	メモリ	なし

【0079】(4-1) パイプライン制御部105は、コンテキスト・バックアップ・メモリ111から命令流3を休眠状態にした要因のLOAD命令の実行PC、オペレーションおよびメモリアクセスアドレスを実行PC\_A55およびオペレーションバッファ53およびロードストアユニット52に復帰させる。命令流3に関する一連の動作を図4に示す。

【0080】(4-2) パイプライン制御部105は、コンテキスト・バックアップ・メモリ111から命令流3の解説PCをFC\_A13に設定する。解説PC\_A15には命令流3のIDを設定しておく。

【0081】(5-1) パイプライン制御部105は、命令流1の休眠状態への遷移で命令フェッチユニット(\_A)や命令解説部\_A14に空きができたこと、命令流3が実行可能状態であり、かつすでにレジスタファイル\_Z109に割り付けられていることから状態管理テーブル115の命令流3の状態を実行可能から実行状態へ遷移させる。それを表3に示す。命令流3に関する一連の動作を図5に示す。

【0082】

【表3】

	状態	レジスタ	命令フェッチユニット
命令流1	休眠	メモリ	なし
命令流2	実行	Y	B
命令流3	実行	Z	A
命令流4	可能	メモリ	なし

【0083】(5-3) パイプライン制御部105は、ロードストアユニット52の実行制御部54に対して、オペレーションバッファ53に設定されたオペレーションやロードストアユニット52に設定されたメモリアクセスアドレスを基に命令流3のLOAD命令を再実行させる。

【0084】(5-4) パイプライン制御部105は、命令流3を実行するため、命令フェッチ制御部\_A12に対してFC\_A13から命令の実行を開始させる。

【0085】(5-5) 命令フェッチ制御部\_A12はFC\_A13のアドレスを起点として、命令フェッチを開始し命令バッファ\_A11に命令を格納する。命令解説部\_A14は命令バッファ\_A11に格納された命令を解説する。以降、定常状態で説明したように命令を処理していく。

【0086】(6-1) パイプライン制御部105は、命令流1の休眠状態への遷移でレジスタファイル\_X107に空

きができたことと、命令流4が実行可能状態であることから命令流4のコンテキストをレジスタファイル\_X107に割り付ける。コンテキスト・バックアップ・メモリ111からレジスタファイル\_X107にデータを転送する。それを表4に示す。

【0087】

【表4】

	状 態	レジスタ	命令フェッチ ユニット
命令流1	休 眠	メモリ	なし
命令流2	実 行	Y	B
命令流3	実 行	Z	A
命令流4	可 能	X	なし

【0088】(6-2) パイプライン制御部105は、命令流4にレジスタファイル\_X107を割り付けたように状態管理テーブル115を更新する。

【0089】(7-1) キャッシュは命令流1が休眠状態へ遷移する要因となったミスヒットのデータをアップデートする。

【0090】(7-2) パイプライン制御部105は、キャッシュにアップデートされたことから、命令流1を休眠状態から実行可能に状態を変更し、状態管理テーブル115を更新する。それを表5に示す。

【0091】

【表5】

	状 態	レジスタ	命令フェッチ ユニット
命令流1	可 能	メモリ	なし
命令流2	実 行	Y	B
命令流3	実 行	Z	A
命令流4	可 能	X	なし

【0092】以上で一連の状態遷移および内部動作を説明したが、それ以降の状態遷移について一例を次に示す。

【0093】命令流2：実行状態→休眠状態

命令流4：実行可能状態→実行状態

命令流1：コンテキスト・バックアップ・メモリ111にレジスタファイル\_Y108を割り付ける

命令流3：実行状態→休眠状態(命令流2：休眠状態→実行可能状態)

命令流1：実行可能→実行状態

以上説明したように、本実施例によれば、

(1) インタロックを発生した命令流を休眠状態(退避)にし、ハードウェア機構を他の命令流に割り付ける、すなわち命令流を切り替えて実行することにより、パイプラインのインタロックおよびロック時間を削減し、機能ユニットの使用効率を向上させ、高性能を実現できる。

【0094】(2) インタロックを発生した命令流を休眠

状態にしてコンテキストをメモリに退避し、新たな命令流のコンテキストを復帰させ、ハードウェア機構を割り付ける、すなわち命令流を切り替えて実行することにより、パイプラインのインタロックおよびロック時間を削減し、機能ユニットの使用効率を向上させ、高性能を実現できる。

【0095】(3) 命令準備部よりも多くのレジスタファイルを設定することにより、インタロックを発生しても、インタロックを発生した命令流のコンテキストをメモリに退避することなく、新たな命令流を実行させることができるので、高速な命令流切り替えができる。

【0096】(4) 命令準備部よりも多くのレジスタファイルとコンテキストを退避するメモリを設けることにより、インタロックを発生した場合、インタロックを発生した命令流のコンテキストはメモリに退避するが、新たな命令流のコンテキストは予めレジスタに準備することができるので、通常の命令と命令流の切り替えがオーバーラップして実行できるので、高速な命令流切り替えができる。

【0097】(5) コンテキストを退避するメモリを同一プロセッサ上に実現することにより、高速な切り替えやロード命令などの外部バスを使用する命令と並列に実行できる。

【0098】(6) 命令流の切り替えの契機となる特定のイベントをキャッシュアクセス時のミスヒットとすることにより、メモリアクセス時のオーバーヘッドを削減できる。

【0099】(7) 特定のイベントが発生したりまたは特定の状態を検出した場合には、その要因となった命令の命令流に割り付けられている命令準備部の実行を停止し、命令準備部の内部状態を初期化することにより、パイプライン制御が容易になる。

【0100】(8) 命令流の実行を停止し退避するコンテキストに先頭PCを含み、再度この命令流を実行する場合には先頭PCから実行を再開することにより、命令流の切り替えの制御が容易になる。

【0101】(9) 命令流の切り替えの契機となる特定のイベントをキャッシュアクセス時のミスヒットとし、キャッシュミスが発生した場合、メモリアクセス命令を打ち切り正常終了させることにより、命令切り替えの制御が容易になる。

【0102】(10) 新たな命令流を命令準備部に割り付け、命令実行を再開する場合、切り替えの要因となり実行を打ち切ったメモリアクセス命令を再実行させることにより、命令流の切り替えの制御が容易になる。

【0103】(11) 命令流の切り替えの契機となる特定のイベントをキャッシュアクセス時のミスヒットとし、命令流のコンテキストを退避する場合にはメモリアクセス命令のPCも一緒に退避することにより、命令再実行が可能となり切り替えの制御が容易になる。

【0104】(12) 命令流の切り替えの契機となる特定のイベントをキャッシュアクセス時のミスヒットとし、命令流のコンテキストを退避する場合にはメモリアクセス命令も一緒に退避することにより、命令再実行が高速に実現でき切り替えの制御も容易になる。

【0105】(13) 命令流の切り替えの契機となる特定のイベントをキャッシュアクセス時のミスヒットとし、命令流のコンテキストを退避する場合にはメモリアクセスアドレスも一緒に退避することにより、命令再実行が高速に実現でき切り替えの制御も容易になる。

【0106】(14) 命令流の切り替えの契機となる特定のイベントをメモリアクセスに関するエラーとすることにより、メモリアクセス時のオーバーヘッドを削減できる。

【0107】なお、本実施例では、レジスタファイルのセット数を3、命令バッファや命令解読部の数を2、そして演算実行部は4種類から構成されているが、これらは本発明の効果や主旨とは関係しないので、構成や個数などについてはなんら制限はしていない。

【0108】本実施例では、再実行するロード命令についてロード命令のPC値およびオペレーションをコンテキスト・バックアップ・メモリに退避しているが、例えばPC値のみを退避しておき、命令フェッチから再実行する方法や、命令を退避しておき解読から再実行する方法などを用いてもかまわない。また、PC値とオペレーションの両方を退避しているのは、例外発生を考慮してのことであり、本発明を限定するものではない。

【0109】本実施例では、説明の都合上ロードストアユニットを使用する例としてロード命令を用いているが、ストア命令についても同様の機能を実現することは可能である。

【0110】本実施例では、情報管理テーブル更新のあとロード命令の再実行という順番で実行しているが、この順番は逆でもよい。この以外の手続きの順番についても本実施例では一例を挙げただけである。

【0111】本実施例では、分岐処理については本発明とは特に関係していないので記述していない。

【0112】本実施例では、切り替えの契機をロード命令としたが、インタロックなどを発生する命令やイベントについても適用できる。

【0113】本実施例では、命令流の切り替え決定後、新たな命令流を設定するまでに、メモリへ退避しているが、レジスタファイルのセット数に余裕があれば、コンテキストのバックアップは新たな命令流を切り替えてから、オーバーラップ実行してもよい。

【0114】本実施例では、プロセッサ上にメモリを設け、休眠状態に入る命令流のレジスタファイルなどのコンテキストをメモリに退避したが、外部メモリに退避してもよい。

【0115】本実施例では、プロセッサ上にメモリを設

け、休眠状態に入る命令流のレジスタファイルなどのコンテキストをメモリに退避したが、オンチップメモリを特に設けず、レジスタファイルに割り付けた命令流で切り替えをしたり、レジスタファイルのコンテキストを切り替えをする場合には外部メモリを使用することも可能である。

【0116】

【発明の効果】以上説明したように、本発明の情報処理装置は、

10 (1) パイプラインのインタロックおよびロック時間を削減し、機能ユニットの使用効率を向上させ、高性能を実現できる。

【0117】(2) 通常の命令と命令流の切り替えをオーバーラップして実行できるので、高速な命令流切り替えができる。

【0118】(3) ロード命令などの外部バスを使用する命令と並列に実行できる。

【0119】(4) メモリアクセス時のオーバーヘッドを削減できる。

20 【0120】(5) パイプライン制御や命令流の切り替え制御が容易になる。

【0121】などの多くのことが実現でき、その実用的効果は大きい。

【図面の簡単な説明】

【図1】本発明の一実施例における情報処理装置の構成図である。

【図2】図1における命令流の状態と状態遷移をさせるイベントの関係を示した状態遷移図である。

30 【図3】図1における命令流が実行状態から休眠状態へ遷移する説明図である。

【図4】図1における命令流が休眠状態から実行可能状態へ遷移する説明図である。

【図5】図1における命令流が実行可能状態から実行状態へ遷移する説明図である。

【図6】図1におけるコンテキスト・バックアップ・メモリのデータ配置図である。

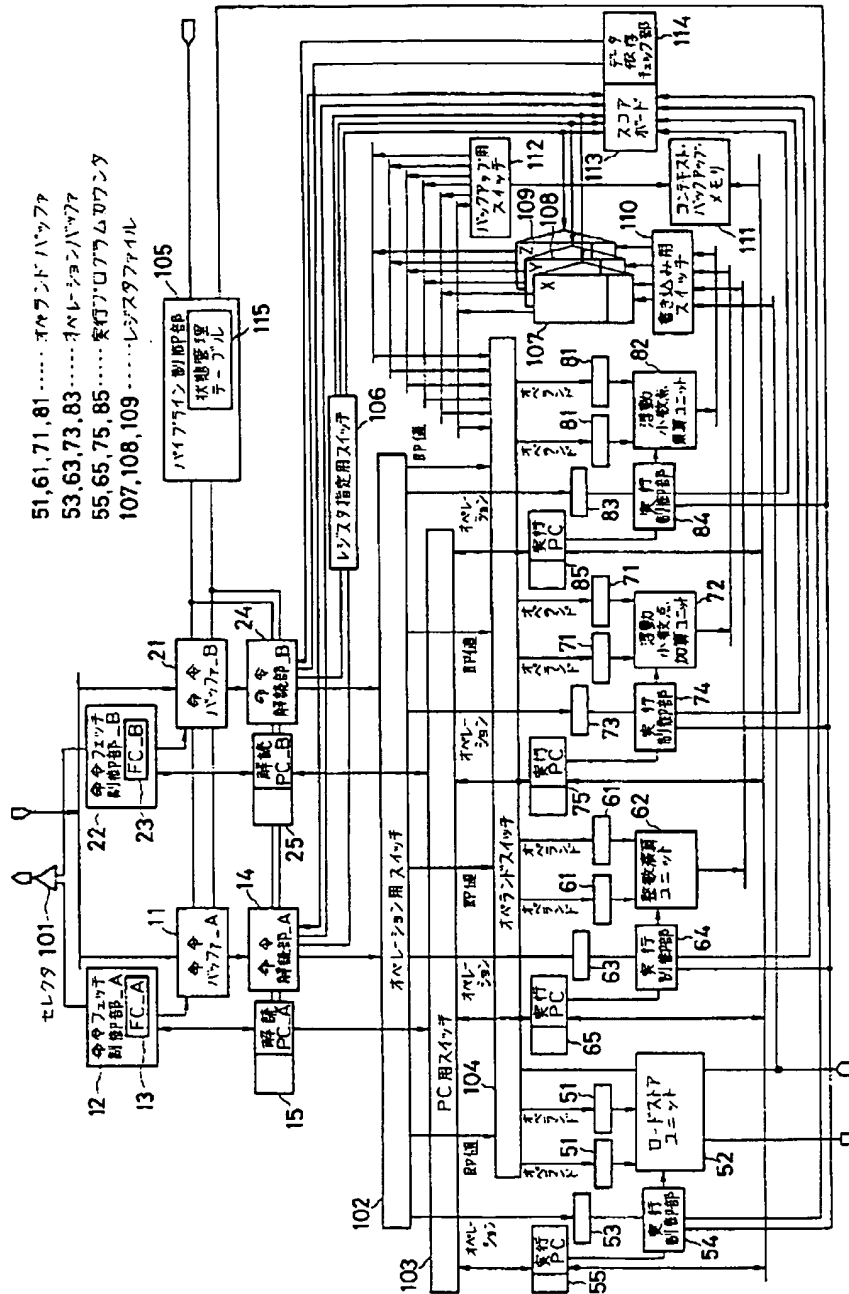
【図7】従来の情報処理装置の構成図である。

【符号の説明】

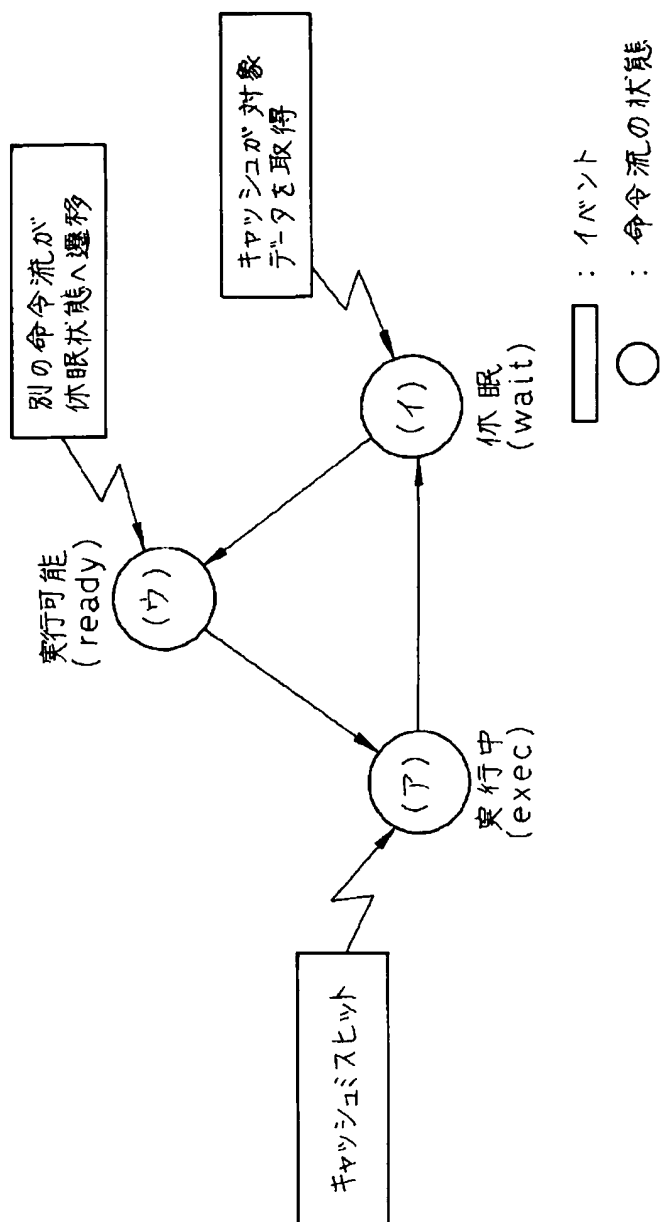
11, 21…命令バッファ、 12, 22…命令フェッチ制御部、 13, 23…フェッチカウンタ(F C)、 14, 24…命令解読部、 15, 25…解読プログラムカウンタ(解読P C)、 51, 61, 71, 81…オペランドバッファ、 52…ロードストアユニット、 62…整数演算ユニット、 72…浮動小数点加算ユニット、 82…浮動小数点乗算ユニット、 53, 63, 73, 83…オペレーションバッファ、 54, 64, 74, 84…実行制御部、 55, 65, 75, 85…実行プログラムカウンタ(実行P C)、 101…セレクタ、 102, 103, 104, 106, 110, 112…スイッチ、 105…パイプライン制御部、 107, 108, 109…レジスタファイル、 111…コンテキスト・バックアップ・メモリ、 1

13…スコアボード、 114…データ依存チェック部、 115 …状態管理テーブル。

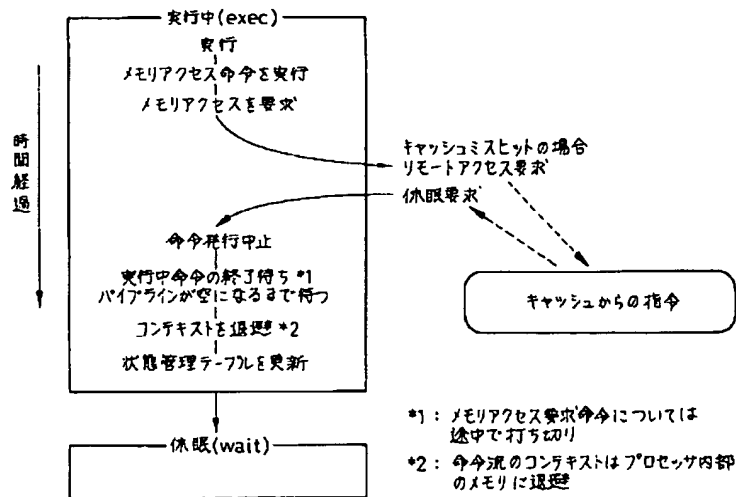
【図1】



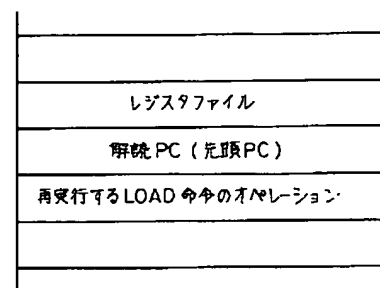
【図2】



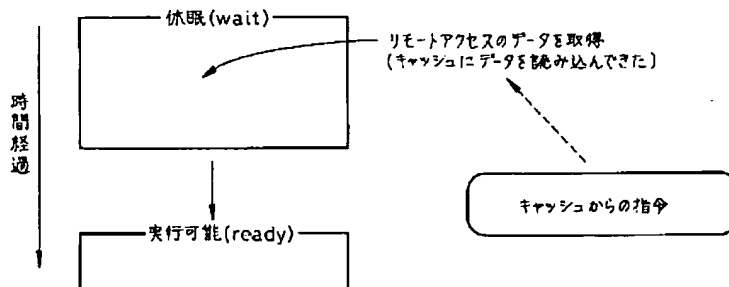
【図3】



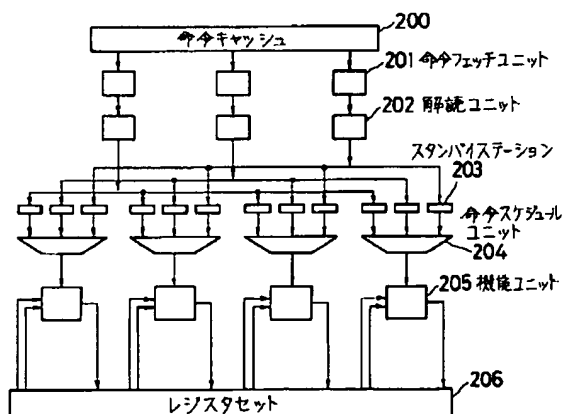
【図6】



【図4】



【図7】





【図5】

